IFW AF
D

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of

Kehsing J. CHOU, et al.

Appln. No. 09/399,696

Confirmation No. 2558

Filed: September 21, 1999

Docket No. A8010 / ST9-99-93

Group Art Unit: 2172

Examiner: Hung Q. PHAM

For: THE ARCHITECTURE AND IMPLEMENTATION OF A DYNAMIC RMI SERVER CONFIGURATION HIERARCHY TO SUPPORT FEDERATED SEARCH AND UPDATE ACROSS HETEROGENEOUS DATASTORES

SUBMISSION OF APPELLANTS' BRIEF ON APPEAL

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith please find an original and two copies of Appellant's Brief on Appeal. A check for the statutory fee of $330.00 is attached. The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account. A duplicate copy of this paper is attached.

Respectfully submitted,

Billy Carter Raulerson

Billy Carter Raulerson
Registration No. 52,156

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Date: August 26, 2004

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of Kehsing J. CHOU, et al. Docket No. A8010 / ST9-99-93

Appln. No. 09/399,696 Group Art Unit: 2172

Confirmation No. 2558 Examiner: Hung Q. PHAM

Filed: September 21, 1999

For: THE ARCHITECTURE AND IMPLEMENTATION OF A DYNAMIC RMI SERVER CONFIGURATION HIERARCHY TO SUPPORT FEDERATED SEARCH AND UPDATE ACROSS HETEROGENEOUS DATASTORES

APPELLANTS' BRIEF ON APPEAL UNDER 37 C.F.R. § 1.192

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.192, Appellants submit that the following comprises the Appellants' Brief on Appeal from the Office Action dated April 7, 2004, wherein claims 1-24 were finally rejected. This Appeal Brief is being filed in triplicate and is accompanied by a Submission which includes the required appeal fee set forth in 37 C.F.R. § 1.17(c). Appellants' Notice of Appeal was filed on July 7, 2004. Therefore, the present Appeal Brief is timely filed.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation ("IBM") of Armonk, New York, the assignee.

II. RELATED APPEALS AND INTERFERENCES

Appellants state that, upon information and belief, Appellants are not aware of any co-pending appeal or interference which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-24 (*see* attached Appendix) are the claims currently on appeal, from the final rejections as set forth in the Office Action dated April 7, 2004.

IV. STATUS OF AMENDMENTS

All amendments filed subsequent to a final Office Action have been entered.

V. SUMMARY OF THE INVENTION

Illustrative embodiments of the present invention relate to a method, an apparatus and an article of manufacture for searching for data in one or more heterogeneous data sources within a computer system (*see* claims 1-24).

For example, illustrative embodiments of the present invention include an architecture and implementation of a dynamic Remote Method Invocation (RMI) server configuration hierarchy to support federated searching and updating across heterogeneous datastores (Appellants' specification: page 2, lines 22-28; page 5, lines 23-27; and Figs. 7-9).

The RMI architecture allows a user to configure a hierarchical grouping of RMI servers either on several different machines or on the same machine to support federated searching and updating across several heterogeneous datastores (Appellants' specification: page 44, lines 22-28). For example, the architecture allows the creation of a flexible tree of RMI servers in which

a new server can be attached or removed dynamically from the configuration, which is advantageous for a federated search environment where the configuration, the number and the type of datastores may change dynamically over time (*Id.*).

In an RMI client/server hierarchy, an RMI server can connect to an unlimited number of datastores, but each RMI server must be connected to at least one datastore (Appellants' specification: page 2, line 29 to page 3, line 3; and Fig. 7). A master RMI server (A) 700 can reference sub-RMI servers (B) 702 and 704 that reside below in the hierarchy (*Id.*). Additionally, another sub-RMI server (C) 706 may be below sub-RMI servers (B) 702 and 704 in the hierarchy (*Id.*).

If an RMI client is searching for the first time for a datastore, the search begins with RMI server (A) 700 (Appellants' specification: page 45, lines 4-7). If the datastore is not found in the RMI server (A) 700, the sub-RMI servers (B and then C) are searched next (*Id.*). If the same RMI client searches for the datastore again, the client searches in the RMI server (A or B or C) where it found the datastore the first time (*Id.*).

When a new RMI server is needed, it can be configured in a different machine (Appellants' specification: page 45, lines 14-20). The additional machine registers or attaches itself to an existing server in the RMI server hierarchy (*Id.*).

Each RMI server is defined with a server type and a maximum number of connections that it can handle (Appellants' specification: page 45, lines 21-24). This information is used by the RMI architecture to perform load balancing and to distribute loads among several servers (*Id.*). The load balancing technique is based, for example, on the percentage of the current load and the maximum load of the server (*Id.*).

By way of example, for two severs (*i.e.*, $Server_A$ and $Server_B$), wherein $Server_A$ can handle 5 loads and $Server_B$ can handle 100 loads, if $Server_A$ is currently handling 4 loads and $Server_B$ is currently handling 20 loads, $Server_A$ is handling a larger percentage of loads relative to its capability (Appellants' specification: page 45, lines 24-28). Therefore, when another request for data is received, $Server_B$ is selected to process the request (*Id.*).

When a federated search request is submitted by a user, the federated datastore will consult the primary node to locate a sever with the proper type and allowable loads (Appellants' specification: page 45, line 29 to page 46, line 3). Then, the federated datastore will direct the search request to the selected server (*Id.*). Once the server capable of providing the requested service is located, any subsequent requests are automatically directed to the selected server, transparent from the user (*Id.*).

VI. ISSUES

The issue on appeal is whether or not claims 1-24 are anticipated by U.S. Patent No. 6,272,488[1] to Chang et al. (hereinafter "Chang"), under 35 U.S.C. § 102(e).

VII. GROUPING OF CLAIMS

The claims do not stand or fall together and arguments for patentability of each group of claims, identified below, are set forth in this brief (*see* Section VIII).

Group 1: claims 1-2, 7-8 and 13-14, each of which stands or fall together.

Group 2: claims 3, 9 and 15, each of which stands or fall together.

[1] U.S. Patent No. 6,272,488 is also assigned to International Business Machines Corporation of Armonk, New York.

Group 3: claims 4-6, 10-12 and 16-18, each of which stands or fall together.

Group 4: claims 19-21, each of which stands or fall together.

Group 5: claims 22-24, each of which stands or fall together.

VIII. ARGUMENTS

1. Claims 1-2, 7-8 and 13-14 are not anticipated by Chang.

Claims 1, 7 and 13, which are all the independent claims pending in the application, recite features that are neither disclosed nor suggested by Chang. Because a reference must disclose each and every feature to anticipate a claim, claims 1, 7 and 13 are not anticipated by Chang.

For example, claim 1 recites, *inter alia*, "selecting a server to process the request based on a load of the server and based on whether the server can satisfy the request for data . . .," which renders the claims of this group patentably distinct from the claims of the other groups. The Examiner alleges Chang discloses these features of claim 1 (*citing* Chang: Fig. 5).

In particular, the Examiner alleges that "the loaded queries into query objects 14-19 as servers will be balanced out by queryable collection 5, and each query object or server will have a load based on the input parameter set up by queryable collection 5 to execute a specific query ..." (*see* Continuation Sheet attached to the Advisory Action dated July 14, 2004).

In Chang, when a user wants to submit a query, he/she starts by creating a specific datastore object 9 to give him/her access to the query processing functions provided by that datastore 9 (Chang: col. 8, lines 40-44). Then, the user calls the "evaluate" method on the datastore 9 and supplies a query string and other parameters (or a query object 13) to process the

query, wherein the result of the query is a queryable collection object 5 that can evaluate further queries (Chang: col. 8, lines 45-52).

Fig. 5 of Chang illustrates, as another way to process a query, creating a query object specific to the type of query language (Chang: col. 8, line 66 to col. 9, line 9). According to Chang, query objects 13 are created using the createQuery() method 40 in the datastore 9 so that the created query object 14-19 will have all the necessary information and can always get help from the datastore 9 in processing the query (Chang: col. 9, lines 1-9).

In no way does Chang disclose "selecting a server to process the request [for data at a federated data source] based on a load of the server", as recited in claim 1. Instead, the queryable collection object 5 relied on by the Examiner is a sequential collection to store the result or scope of a query, which can be further queried (Chang: col. 6, lines 61-65; and col. 7, lines 13-14). Furthermore, the queryable collection 5 can be provided as an input parameter to an execute method 41 to limit the scope of a query (Chang: col. 9, lines 6-9).

The scope of any particular query, however, does not correspond to the overall load of a server. Indeed, a server may be processing many queries, each with varying scopes. Consequently, Chang does not disclose selecting a server based on the load of the server.

In view of the above, it is respectfully submitted that claim 1 is not anticipated by Chang. Claims 7 and 13 recite features similar to claim 1 and thus are not anticipated by Chang based on a rationale analogous to that set forth above for claim 1. Consequently, claims 2, 8 and 14 are not anticipated by Chang at least by virtue of their dependency.

2. Claims 3, 9 and 15 are not anticipated by Chang.

Claims 3, 9 and 15 are not anticipated by Chang at least by virtue of their dependency (*see* Section VIII(1)), as well as the additional features recited therein. For example, claim 3 recites the operation of "forwarding additional requests for similar data to the selected server," which renders the claims of this group patentably distinct from the claims of the other groups. The Examiner alleges that Chang discloses these features of claim 3 (*see* Final Office Action: page 6, *citing* Chang: col. 8, lines 53-57).

To the contrary, Chang describes using a queryable collection object to define an input scope for the execute method in query objects (Chang: col. 8, lines 53-57). Limiting the scope of a query does not correspond to "forwarding additional requests for similar data to the selected server," as recited in claim 3.

In view of the above, it is respectfully submitted that claim 3 is not anticipated by Chang. Claims 9 and 15 recite features similar to claim 3 and thus are not anticipated by Chang based on a rationale analogous to that set forth above for claim 3.

3. Claims 4-6, 10-12 and 16-18 are not anticipated by Chang.

Claims 4-6, 10-12 and 16-18 are not anticipated by Chang at least by virtue of their dependency (*see* Section VIII(1)), as well as the additional features recited therein. For example, claim 4 recites that "the server is within a server hierarchy," which renders the claims of this group patentably distinct from the claims of the other groups. The Examiner alleges that Chang discloses these features of claim 4 (*see* Final Office Action: page 6, *citing* Chang: Fig. 2).

Fig. 2 of Chang illustrates a diagram of the class hierarchy for query classes, wherein there is a query class for each type of query (Chang: col. 7, lines 27-40). As noted in Chang, a

"class" defines the implementation of a particular kind of object, the variables and methods it uses, and the parent class it belongs to, as used in object-oriented programming (Chang: col. 3, lines 5-13). A class does not correspond to a server. Thus, Chang fails to disclose or suggest any server hierarchy.

In view of the above, it is respectfully submitted that claim 4 is not anticipated by Chang. Claims 10 and 16 recite features similar to claim 4 and thus are not anticipated by Chang based on a rationale analogous to that set forth above for claim 4. Consequently, claims 5-6, 11-12 and 17-18 are not anticipated by Chang at least by virtue of their dependency.

4. Claims 19-21 are not anticipated by Chang.

Claims 19-21 are not anticipated by Chang at least by virtue of their dependency (*see* Section VIII(1)), as well as the additional features recited therein. For example, claim 19 recites that "said load of the server is based on at least the ratio of a current load of the server and a maximum load of the server," which renders the claims of this group patentably distinct from the claims of the other groups. The Examiner alleges that Chang discloses these features of claim 19 (*see* Final Office Action: page 7, *citing* Chang: col. 31, lines 9-44).

To the contrary, Chang describes a DL (digital library) parametric query string (Chang: col. 31, lines 9-44). The parametric query string can accept parameters supplied by users, including the name of an index class to be searched, the maximum number of results to be returned and a conditional expression (*Id.*).

As noted in Chang, a parametric query is a query that requires an exact match on the condition specified in the query predicate and the data values stored in the datastore (Chang: col.

5, lines 28-30). Thus, a parametric query string does not correspond to a load of a server, let alone a load based on the ratio of a current load of the server to a maximum load of the server.

In view of the above, it is respectfully submitted that claim 19 is not anticipated by Chang. Claims 20 and 21 recite features similar to claim 19 and thus are not anticipated by Chang based on a rationale analogous to that set forth above for claim 19.

5. Claims 22-24 are not anticipated by Chang.

Claims 22-24 are not anticipated by Chang at least by virtue of their dependency (*see* Section VIII(1)), as well as the additional features recited therein. For example, claim 22 recites that "the server is a Remote Method Invocation server," which renders the claims of this group patentably distinct from the claims of the other groups. The Examiner alleges that Chang discloses these features of claim 22 (*see* Final Office Action: page 7, *citing* Chang: col. 63, lines 7-18).

Chang describes a preferred implementation in an object-oriented language such as the JAVA or C++ languages (Chang: col. 63, lines 7-18). The Examiner sites a web page in alleging that "Java has RMI server and client to support seamless remote invocation on objects in different virtual machine by default" (Final Office Action: page 7). The Examiner then jumps to the unsupported and erroneous conclusion that the remote query objects 14-19 of Chang must use an RMI model to implement the technique to have a Remote Method Invocation server (*Id.*).

Chang relates to managing the results of federated searches across heterogeneous datastores with a federated collection object (Chang: Abstract). As noted above, such an object may be implemented in an object-oriented programming language such as JAVA, which happens

to support remote procedure calls (*i.e.*, RMI). The objects of Chang, however, are not servers, let alone RMI servers.

In view of the above, it is respectfully submitted that claim 22 is not anticipated by Chang. Claims 23 and 24 recite features similar to claim 22 and thus are not anticipated by Chang based on a rationale analogous to that set forth above for claim 22.

IX. CONCLUSION

Appellants respectfully request the members of the Board to reverse the rejections of the appealed claims and to find each of the claims allowable as defining subject matter that is patentable over the art of record.

The present Brief on Appeal is being filed in triplicate. Unless a check is submitted herewith for the fee required under 37 C.F.R. §1.192(a) and 1.17(c), please charge said fee to Deposit Account No. 19-4880.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,

Billy Carter Raulerson

Billy Carter Raulerson
Registration No. 52,156

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Date: August 26, 2004

APPENDIX

CLAIMS 1-24 ON APPEAL:

1. A method for searching for data in one or more heterogeneous data sources within a computer system, the method comprising the steps of:

receiving a request for data at a federated data source; and

selecting a server to process the request based on a load of the server and based on whether the server can satisfy the request for data, said server connected to one or more heterogeneous datastores.

2. The method of claim 1, further comprising forwarding the request to the selected server.

3. The method of claim 2, further comprising forwarding additional requests for similar data to the selected server.

4. The method of claim 1, wherein the server is within a server hierarchy.

5. The method of claim 4, further comprising, upon receiving a request to add another server, connecting the server to an existing server in the server hierarchy based on a number of connections of the existing server.

6. The method of claim 4, further comprising, upon receiving a request to delete an existing server in the hierarchy, deleting that server.

7. An apparatus for searching for data in one or more heterogeneous data sources, comprising:

a computer system having one or more heterogeneous data sources; and

one or more computer programs, performed by the computer system, for receiving a request for data at a federated data source and selecting a server to process the request based on a load of the server and based on whether the server can satisfy the request for data, said server connected to one or more heterogeneous datastores.

8. The apparatus of claim 7, further comprising forwarding the request to the selected server.

9. The apparatus of claim 8, further comprising forwarding additional requests for similar data to the selected server.

10. The apparatus of claim 7, wherein the server is within a server hierarchy.

11. The apparatus of claim 10, further comprising, upon receiving a request to add another server, connecting the server to an existing server in the server hierarchy based on a number of connections of the existing server.

12. The apparatus of claim 10, further comprising, upon receiving a request to delete an existing server in the hierarchy, deleting that server.

13. An article of manufacture comprising a program storage medium readable by a computer system and embodying one or more instructions executable by the computer system to perform method steps for searching for data in one or more heterogeneous data sources within a computer system, the method comprising the steps of:

receiving a request for data at a federated data source; and

selecting a server to process the request based on a load of the server and based on whether the server can satisfy the request for data, said server connected to one or more heterogeneous datastores.

14. The article of manufacture of claim 13, further comprising forwarding the request to the selected server.

15. The article of manufacture of claim 14, further comprising forwarding additional requests for similar data to the selected server.

16. The article of manufacture of claim 13, wherein the server is within a server hierarchy.

17. The article of manufacture of claim 16, further comprising, upon receiving a request to add another server, connecting the server to an existing server in the server hierarchy based on a number of connections of the existing server.

18. The article of manufacture of claim 16, further comprising, upon receiving a request to delete an existing server in the hierarchy, deleting that server.

19. The method of claim 1, wherein said load of the server is based on at least the ratio of a current load of the server and a maximum load of the server.

20. The apparatus of claim 7, wherein said load of the server is based on at least the ratio of a current load of the server and a maximum load of the server.

21. The article of manufacture of claim 13, wherein said load of the server is based on at least the ratio of a current load of the server and a maximum load of the server.

22. The method of claim 1, wherein the server is a Remote Method Invocation server.

23. The apparatus of claim 7, wherein the server is a Remote Method Invocation server.

24. The article of manufacture of claim 13, wherein the server is a Remote Method Invocation server.